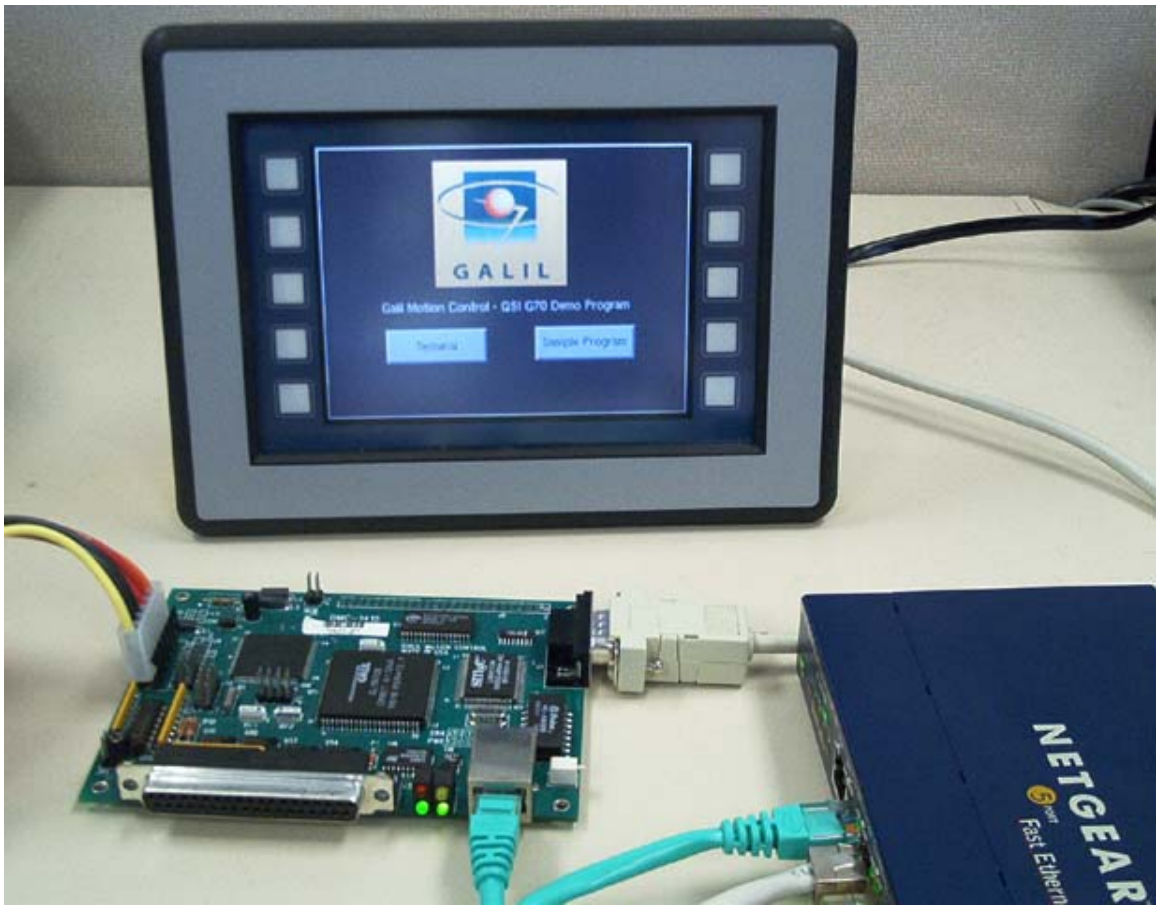
Application Note #5442

Connecting the QSI QTERM-G124 Operator Interface to Galil Controllers

Introduction

This application note explains how to interface any Galil stand-alone controller to the QTERM-G124 Operator Interface. (*Note: the G124 is functionally the same as the G70 offered from QSI, however the G124 denotes usage with Galil Controllers*). The QSI QTERM-G124 is a 320x240 (256 color or grayscale) LCD touch-screen display with Ethernet 10Base-T and RS-232 capability. Using a software program called Qlarity Foundry, users can program the terminal using an object oriented programming language with BASIC-like commands and standard programming structures such as IF/THEN and WHILE, FOR-NEXT looping. For complete specifications on the G124 and a complete version of the free Qlarity Foundry software package, please visit the website http://www.qsicorp.com/.



Galil Motion Control, Inc. • 3750 Atherton Road • Rocklin, CA 95765 USA • 800-377-6329 • Ph: 916-626-0101 • Fax: 916-626-0102 • www.galilmc.com

In the above photo, the controller being used is a DMC-1425, but any stand-alone controller from Galil will allow the same connection. One major benefit of the QSI terminal is the ability to connect over Ethernet. This note will outline the procedure of connecting from the controller to the G124 over both Ethernet and RS-232. A downloadable version of the demo programs written in Qlarity and used to connect to the controller are available on the Galil site- http://www.galilmc.com/ftp/pub/appnotes/qsi .

Procedure

I. First, you must download and install the latest Qlarity Foundry software package from the QSI website at: http://www.qsicorp.com/qterm_g70-g75_downloads.html#software. Once installed, you will need to download the file galil_demo.zip from http://www.galilmc.com/ftp/pub/appnotes/qsi which includes the Galil Object Library (galil_mc.lib) and sample program (galil_demo.qly). After downloading galil_demo.zip, unzip the files into a folder located on your hard drive: C:\Program Files\QSI Corporation\Qlarity Foundry\Galil. The object library includes a communication object as well as many objects for executing and displaying commands on the Galil controller. The demo program demonstrates many of the features of the QSI terminal and the Galil objects.

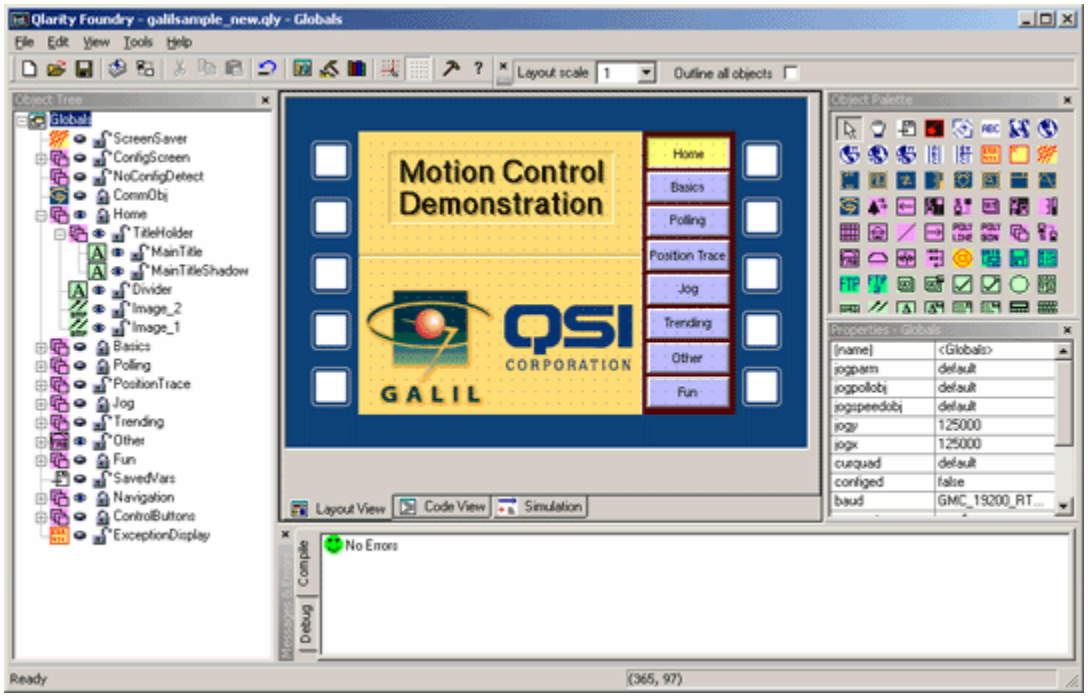
II. It will probably help to read the G70 User Manual and Qlarity Programming Guide to familiarize the user to the programming environment, as well as going through some of the tutorial programs. Once there is an understanding of the basics of the Qlarity environment, it becomes much easier to see how things work. Please note a few key features that are extremely helpful while programming. The first one is the Simulation mode with "virtual" serial ports. This allows the user to write the code without having a physical terminal present. The simulator emulates the hardware and allows the user to see what happens while communicating via the Serial port. Another helpful tool is the Object Tree – which allows the user to view all the screens, buttons, and objects in the code as well as the programming hierarchy in a familiar “Tree” format. Much like a Visual Basic program, objects are placed on a screen with their own properties that are easily changed via the Properties Window. Also like V.B, is the ability to layout the screen visually, and then go "behind" the objects to write the code.

III. The next step is to set up the Galil controller so that communication is possible. If not done already, install the Galil SmartTerm software. If ethernet communication between the G124 and controller is desired, then connect straight through ethernet cables from the hub to: the PC, the controller, and the G124 unit. Then use the SmartTerm registration utility to "Find Ethernet Controllers" – pick the Galil controller and assign it an IP address (in this case- 10.0.51.201)[1]. Connect to the Galil controller over Ethernet using the standard Galil Terminal window. Enter the command "BN" to burn in the IP address and then "TH" (Tell Handle) to make sure that the only connection on the controller is on Handle A and is between the PC and the controller card.

Galil Motion Control, Inc. • 3750 Atherton Road • Rocklin, CA 95765 USA • 800-377-6329 • Ph: 916-626-0101 • Fax: 916-626-0102 • www.galilmc.com

If serial (RS-232) communication is desired, then simply connect a **"Null Modem"** serial cable from the QSI G124 to the controller (this is **not** the same as the "Straight-Through" cable that is required when connecting from the PC to the controller). Be sure that the baud rate is set and hardware handshaking is enabled on the controller and in the Qlarity communication object.

IV. Next, use Qlarity Foundry to open the downloaded demo program called "galil_demo.qly" that should now be stored on your hard-drive. A screen similar to this one should now be up on the PC:



The left hand side shows the "Object Tree" which displays a hierarchy of screens and the objects contained in them. Click on the "+" to expand the selection, and click on the eyeball to either Show or Hide the item. The center section contains the *Layout View*, the *Code View* and the *Simulation* screens. The *Layout View* is where objects can be placed on the screen creating the G124's Graphical Interface. The *Code View* is what sits behind the Graphical Interface and allows the user to write further code to handle events that are triggered. Lastly, the *Simulation* screen is where the code is compiled and run in a "virtual" setup that emulates what the G124 will display including sending and receiving messages over the serial ports and returning error messages. The bottom of the screen contains both the error messages and the compiler results.

The right hand side of Qlarity Foundry contains the "Object Palette" and the "Object Properties" windows. Click on an object in the "Palette" window and draw it in the *Layout View* to create the graphical interface. Then click on the object to display its properties in the "Properties" window.

Galil Motion Control, Inc. • 3750 Atherton Road • Rocklin, CA 95765 USA • 800-377-6329 • Ph: 916-626-0101 • Fax: 916-626-0102 • www.galilmc.com

V. Next, connect a standard ethernet "patch" cable from the hub to the G124 unit, or if preferred, connect a serial cable from the G124 Terminal to the PC (see G124 hardware manual for pinout of connector). To put the G124 in a mode that is ready to receive an application program from the PC, follow these steps:

a. First, hold a finger on the top left corner of the G124 as power is applied. A Setup screen will appear and the user will need to hit SELECT to change the "download" option from None to "App(/FW)". Also, if the user does not want to burn the program into the G124's non-volatile memory, then change "App Mode" to "*Develop*" [2].
b. Then go to the NETWORK settings and change the IP address to 10.0.51.200.[1]
c. Next, select Save and Exit from the DONE option – this puts the touch screen into a mode that it is ready to receive an application.

When ready, use Qlarity Foundry to download the application onto the G124, click on "File – Download Application". From the Download window, select either Serial and enter your parameters, or Ethernet and enter the IP address you set on the G124 (10.0.51.200[1]). The preferred method is ethernet since it is a much faster download. When the program comes up on the G124 screen, use the navigation buttons to choose different screens. It should automatically bring up a window that allows you to choose the desired method of communication between the Controller and the G124 unit. Enter either the IP address of the controller or the serial parameters of the controller you are connected to directly on the Touchscreen interface. Once connected, the user can use the navigation buttons on the right hand side to select different screens. Each screen has an example of objects that can be used in a User application.

VI. When ready to start a ***new*** application using the Galil object library without using the galil_demo.qly - follow these steps:

1. Start a new 'programmable' workspace in Qlarity Foundry. Select "Edit Libraries" from the "Edit" menu. Click on the button labeled "Add existing library." On the dialog box that appears, click "Add Other Library". You can then browse for and select the "galil_mc.lib" file that was downloaded.

2. This will add the Galil library to your workspace, and the Galil objects should now appear in the object palette (they are the blue and orange ones). To begin your workspace, start by selecting the "GalilCommunication" object and drawing it on the *Layout View* screen. The first thing that needs to be set for proper communication is the "CommObj" properties. Use the object tree to click on this item and set the property **commmethod** to your preferred communication method – ethernet (tcp/ip), ethernet (udp), or serial. Then set the properties for **ipaddress[1], serialformat,** and **serialport** accordingly. For this example, the commmethod is set to *gmc_ethernettcp* and the ipaddress field is set to 10.0.51.200[1].

Galil Motion Control, Inc. • 3750 Atherton Road • Rocklin, CA 95765 USA • 800-377-6329 • Ph: 916-626-0101 • Fax: 916-626-0102 • www.galilmc.com

3. Usually, the first step in creating a program is to click on the "Screen" object and draw it on the screen – this will be the first "page" of your program. In this example, we will place a few different objects on the screen just to show how it is done.

4. Polling objects are very useful for displaying controller info on the screen. Use the properties window to specify what data you need to poll for. Enter the Galil command into the *command* property and then enter the poll interval in milliseconds. Any command can be given such as "MG_TVX" for the Velocity of the X axis, or just "Temp=" that will constantly return what value is stored in the Galil variable Temp. Place one polling window on the screen and set its command property to "TPX". Then place a label object next to it and put "Motor Position" as the *caption*.

5. Add a Simple Terminal to be able to send commands and receive responses. Included in this object is a keyboard that enables the user to type in custom commands to send to the controller at run-time.

6. Put in some command buttons with the caption of "Jog X Motor" and "Stop X Motor" using the Properties window. Also, include the commands "JGX=1000;BGX" and "STX" in the *command* property, for each one respectively.

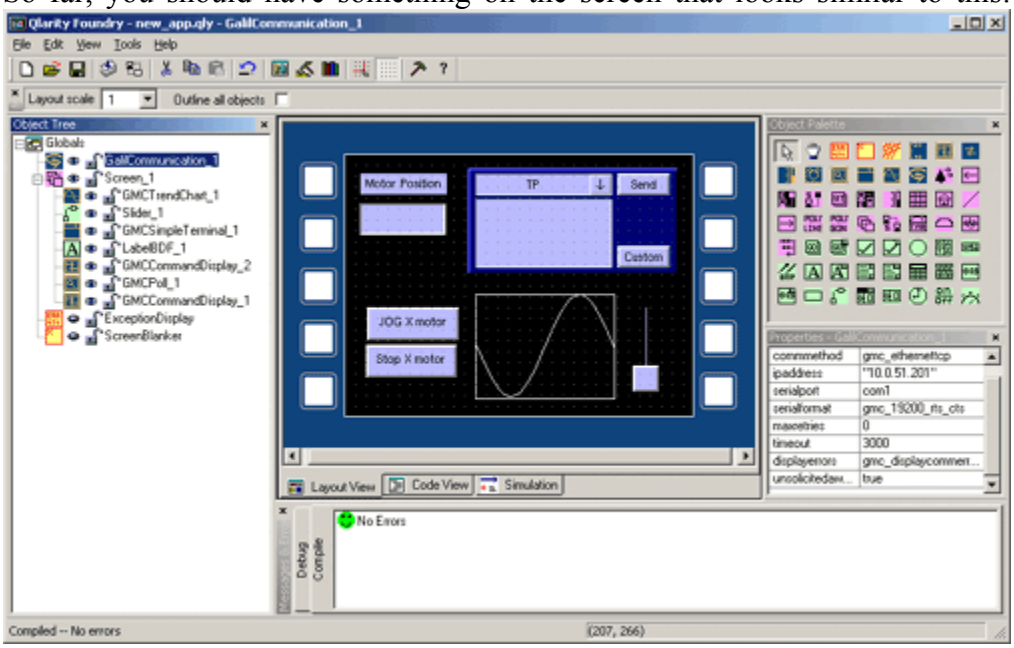
7. Include a TrendChart and set the *inputtype* and *inputsource* properties to capture the correct data. For this example, set *usecustomcommand* to "true" and type in "MG_TVX" under the *customcommand* property. Also, set the *pollinterval* to 500 and set the range from 0 – 2000.

8. Lastly, for advanced users who wish to take advantage of objects that are not galil-specific such as a "Slider" object - , it is easy to use them to send Galil commands to the controller. Create a Slider object called slider_1. Select it, and enter Code View. Near the top of the display is a drop down list titled "*Events / Overrides*." Select the entry called "Change" from the drop down list. A function outline will appear in the visible code. Fill out the function as follows:

```
func Change()
    GalilCommunication_1.SendAsciiCommandStream("JGX=" + str(slider_1.value) +
";BGX;", default, 0)
endfunc
```

This sends down the value of the slider as a Jog command anytime the Slider position is changed. Change the slider properties to have a range of 0-2000.

9. So far, you should have something on the screen that looks similar to this:



Once finished, download the application to the G124 unit as shown previously. You should be able to easily create a custom User Interface with multiple screens using the pre-built objects similar to the galil_demo.qly program. You can attempt to test your workspace in Qlarity Foundry's Simulation View. Be aware that the simulation view will only show virtual communication over the serial ports and no ethernet communication at this time. To configure Simulation View to use your PC serial port(s) to emulate the ports on the G124, select "Settings" from the "View" menu. Select the "Simulation" tab and configure the default values for the serial ports. Be sure to select a PC serial port to map to the serial port on your G124.

TIPS & TRICKS

1 The user can figure out the IP address of the PC by opening a DOS prompt and typing in "ipconfig" (without quotes). If the Subnet Mask is 255.255.255.0 – it means the PC will only communicate with devices that have the same IP address as the PC in the first 3 fields (ie: where the 255's are). So, in this example, the PC's address is 10.0.51.100 – so the address 10.0.51.201 is assigned to the controller and 10.0.51.200 is assigned to the G124 so that communication is possible. This will be different in each system – based on the IP address of the computer. Either the PC's IP address will need to be changed or the IP addresses that are given to the controller and G124 will. Contact your system administrator if you have questions about appropriate IP addresses to assign to your controller and / or G124 terminal.

2 If "App Mode"' is set to "develop", then the terminal will expect a new application to be sent to it each time the terminal is powered on. Once development has been completed, simply set the "App Mode" back to

"Normal," and set "Download" to "App/(FW)" to download an application and save it to the terminal's non-volitile memory. When "App Mode" is set to normal, each time the terminal is powered on, the terminal will run the application saved in its non-volitile memory.

Appendix A– Pre-Built Object Descriptions

'***
Author: Jeremy Richards
Date: 01 Nov 2002
Copyright: 2002 QSI Corporation. Permission to modify and
distribute this object is granted provided this
copyright notice remains in tact.
'***

Object: GalilCommunication
Description: Galil motion controller communications object

This object forms the core of the Galil library. This object controls all interaction between the motion controller and the QTERM-G70. It is capable of communicating with a motion controller either serially or via ethernet. To communicate with a motion controller, create an instance of this object and set up its properties as appropriate. Then create other GMC objects and set thier commObject property to refer to this object. You can communicate with multiple motion controllers by creating one GalilCommunication object per controller. There is a limit of one controller per serial port on the G70 and a limit of 8 ethernet controllers that can be used at once.

Object: GMCScaleObject
Description: Assistant scale display for GMC objects

Object: GMCCommand
Description: Executes a command (or series of commands)
on the motion controller when you push this button
When the command(s) complete execution, the results
are optionally displayed.

This object provides a simple interface to execute an arbitray command (or series of commands) on your Galil Motion Controller. This object appears as a push button which when pushed, executes the commands defined by the command property, then displays the results in the button.

Object: GMCGauge
Description: a vertical/horizontal bar gauge for visual representation

This object implements a flexible bar graph to graphically indicate the value of a paramater on the motion controller The object supports user-defined start and endpoints and an optional color gradient across the length of the bar.

Object: GMCPathTrace
Description: Galil motion controller Path tracing object

This object implements a type of 2D graph that can be used to plot the recent positions of the motor. To use this object, simply set up its properties, especially the startRangeHorz, endRangeHorz, startRangeVert, endRangeVert, commObject, inputTypeHorz, inputSourceHorz, inputTypeVert, inputSourceVert, and pollInterval properties. Generally you would be interested in polling the motor position along two of the motor axis (such as the X and the Y axis). This object is robust object and can consume a lot of the processing power availble on the QTERM-G70. As such, you should minimize the use of other polling type objects on the same screen as you place this GMCPathTrace object. You can also improve performance, if necessary, by increasing pollInterval, and decreasing numTracePoints.

Object: GMCPoll
Description: Galil motion controller polling object

This object polls the controller with an arbitrary command at regular intervals. It then displays the controller's response. It is useful for displaying values that will change over time, such as the motor position. Many properties are provided which contorol the appearance of this object, the polling frequency and what is being polled. Note: To enhance performance, this object will only poll the controller when it is visible (enabled).

Object: GMCSimpleTerminal
Description: A simple terminal for entering interactive commands to the controller.

This object is a simple terminal window used to send arbitray commands to the controller. You can use this object to compose commands to send to the motion controller using a simple on-screen keypad. Recent commands are saved so that they may be resent. The results of the commands are displayed in the terminal window at the bottom of the object.

Object: TrendChart
Description: This object draws a graph of data received

Use Inputtype and Inputsource properties to choose one of the standard polling displays – or choose a custom command to graph.